REMARKS

Claims 1–20 are pending in the application with claims 1, 13, and 19 being the independent claims. No claims are amended, newly added, or canceled without prejudice to or disclaimer of the subject matter therein. Reconsideration of presently pending claims 1–20 is respectfully requested in light of the above amendments and the following remarks.

Allowable Subject Matter

Noted with appreciation is the indication in the Office Action that claims 10 and 17 are directed at allowable subject matter and would be allowed if rewritten in independent form. Office Action, Page 7. Claims 10 and 17 depend from rejected independent claims 1 and 13 respectively. However, for reasons set forth below, it is believed that independent claims 1 and 13 are allowable. Thus, it is believed to be unnecessary to separately place claims 10 and 17 in independent form at this time.

Rejections under 35 U.S.C. § 103

Claims 1–2, 7, 13, and 19 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Dangelo et al. (U.S. Patent No. 5,493,508, hereinafter referred to as "Dangelo") in view of Beausang et al. (U.S. Patent No. 5,703,789, hereinafter referred to as "Beausang"). In addition, claims 3–6, 8–9, 11, 15–16, and 20 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Dangelo in view of Beausang and further in view of Zizzo (U.S. Patent No. 6,578,174). Claims 12, 14, and 18 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Dangelo in view of Beausang in view of Zizzo and further in view of Watanabe et al. (U.S. Patent No. 6,157,947). Applicants respectfully traverse these rejections on the grounds that these references are defective in establishing a *prima facie* case of obviousness.

In *KSR Int'l. Co. v. Teleflex Inc.*, 127 S. Ct. 1727, 1739 (2007), the Court stated that:

> [A] patent composed of several elements **is not proved obvious merely by demonstrating that each of its elements was, independently, known in the prior art.** Although common sense directs one to look with care at a patent application that claims as innovation the combination of two known devices according to their established functions, it can be important to **identify a reason that would have prompted a person of ordinary skill in the relevant field to combine the elements in the way the claimed new invention does.** This is so because inventions in most, if not all, instances rely upon building blocks long since uncovered, and claimed discoveries almost of necessity will be

> combinations of what, in some sense, is already known." *Id.* at 1741 (emphasis added).

As the PTO recognizes in MPEP § 2142:

> The examiner bears the initial burden of factually supporting any *prima facie* conclusion of obviousness. If the examiner does not produce a *prima facie* case, the applicant is under no obligation to submit evidence of nonobviousness.

It is submitted that, in the present case, a *prima facie* case of obviousness does not exist for the claims for the following, mutually exclusive, reasons.

Independent Claim 1

1. The Examiner has not shown that all words in the claim have been considered.

MPEP § 2143.03 states that "[a]ll words in a claim must be considered in judging the patentability of that claim against the prior art." (quoting *In re Wilson*, 424 F.2d 1382, 1385 (CCPA 1970)). However, in the present matter, the Examiner has not shown that all words in the claim have been considered. For example, independent claim 1 requires "prompting, via a multi-compiler interface, for a selection of a first memory compiler unit from a plurality of memory compiler units."

The Office Action notes that "Dangelo doesn't expressly disclose wherein prompting, via a multi-compiler interface, from a selection of a first memory compiler unit from a plurality of memory compiler units." Office Action, Pages 2–3. However, the Office Action contends that Beausang "discloses a multiple compiler environment which utilizes a first compiler (615) which performs a process on the generics net list of the IC (13:20-30), and also a second compiler (625) is also selected to process to optimize the results (13:50-60)." Office Action, Page 3. Applicants respectfully disagree.

The Office Action's references to column 13, lines 20–30, 50–60 provide:

> This generic compiler 615 transforms the HDL description 605 into a technology independent netlist 620 that is more readily recognized by the TR compiler 625. Block 615 performs a process on the input netlist 620 to generate a technology independent or generic netlist of the IC layout by interfacing with a synthetic library or "designware" library. Technology independent netlist 620 is composed of logical primitives and operators of the IC layout but the components described therein contain no structure. The generic compile process 615 and resultant output 620 are well known in the art.

* * *

> Referring to FIG. 8, by performing the above processes, the TR compiler 625 of the present invention is able to better optimize for the eventual construction and completion of the DFT circuitry. In this way, it is more likely that the resultant test circuit design will meet constraints 610. The output of the TR compiler 625 of the present invention is a non-scannable technology dependent netlist 630 that comprises scannable memory cells with loopback connections 440. Although called "nonscannable" because of the loopback connections, netlist 630 is nevertheless a fully scanned netlist in that the TR compiler 625 replaced each HDL specified or inferred sequential cell (e.g., non-scan cell) by an equivalent scannable cell.

(emphasis added). Though Beausang teaches "generic compiler 615" and "TR compiler 625," Beausang clearly fails to teach "prompting . . . for a selection of a first memory compiler unit from a plurality of memory compiler units" as is recited in claim 1. (emphasis added). Nowhere does Beausang teach, disclose, or suggest prompting a selection of either the generic compiler 615 or TR compiler 625. To the contrary, Beausang teaches that both generic compiler 615 and TR compiler 625 are utilized: generic compiler 615 "transforms the HDL description 605 into a technology independent netlist 620 that is more readily recognized by the TR compiler 625," and TR compiler outputs "a non-scannable technology dependent netlist 630 that comprises scannable memory cells with loopback connections 440."

Further, Beausang fails to teach prompting a selection of a first memory compiler unit from a plurality of memory compiler units via a multi-compiler interface. Applicants respectfully submit that Beausang's teaching, "[t]he HDL description 605 can be stored in a computer memory unit (e.g., unit 102 or 104) and is fed into an optional generic compiler logic block 615" (column 13, lines 16–20), is not equivalent to prompting a selection via a multi-compiler interface.

Thus, for this independent reason alone, the Examiner's burden of factually supporting a *prima facie* case of obviousness has clearly not been met, and the rejection under 35 U.S.C. § 103 should be withdrawn.

2. The Examiner has not shown how the elements being combined are performing their known or established function.

KSR teaches that when combining elements from different references, it is important to determine whether the element is performing "the same function it had been known to perform." *KSR* at 1740. It is clear that the generic compiler 615 and TR compiler 625 of Beausang should

not be combined with the system of Dangelo because the known function of compilers 615, 625, specifically TR compiler 625, is changed.

More particularly, Beausang teaches that a need exists for a "system that can reduce the time required to perform circuit synthesis while providing effective DFT processes" and "to provide the compile process with information relating to the impact of testability cells and resources to the overall design so that this information can be accounted for during the initial compilation phase." (column 4, lines 16–18, 26–30). Due to those needs, the primary function of TR compiler 625 is:

> [T]he TR compiler optimizes with increased efficiency for the added test resources so that predetermined performance and design related constraints of the mission mode circuitry are maintained after addition of the test resources. The TR compiler translates generic sequential cells into technology dependent non-scan sequential cells. In the TR compiler, during replacement, scannable memory cells are used in place of these non-scan memory cells specified within the mission mode design and therefore the TR compiler is informed of the characteristics of the scannable memory cells and their links. For test, the scannable memory [cells] are chained to each other to form scan chains. To account for this chaining during compile, the TR compiler provides output driven loopback connections to simulate the electrical characteristics of this chaining during compile. In the above implementation, the TR compiler can efficiently provide translation of an TR description having test implementations into a gate level netlist. With the addition of certain information regarding the test implementation (e.g., scan replacement is done and loopback connections are added), the TR compiler of the present invention better optimizes the overall design for the addition of the test resources which are linked together for testability.

(column 4, lines 48–68; column 5, lines 1–3) (emphasis added). Clearly, the TR compiler provides an important purpose in the Beausang patent, "optimiz[ing] the eventual construction and completion of the DFT circuitry." (column 13, lines 52–53).

If, as the Office Action suggests, Beausang's generic compiler 615 and, more specifically, Beausang's TR compiler 625 is "a first compiler" and "a second compiler," the TR compiler's purpose as an optimization feature is destroyed since "prompting, via a multi-compiler interface, for a selection of a first memory compiler unit from a plurality of memory compiler units" would allow selecting either the generic compiler 615 or TR compiler 625. Accordingly, the TR compiler 625 may not be selected, eliminating the optimization process of the Beausang patent.

Thus, since this modification of the Beausang patent clearly destroys the purpose or function of the TR compiler 625, one of ordinary skill in the art would not have found a reason to make the claimed modification. For this reason alone, the Examiner's burden of factually supporting a *prima facie* case of obviousness has clearly not been met, and the rejection under 35 U.S.C. § 103 should be withdrawn.

Independent Claim 13

Independent claim 13 was also rejected as being unpatentable over Dangelo in view of Beausang. It is clear that the Dangelo and Beausang references are not properly combinable since if combined, neither reference teaches prompting selection of a memory compiler unit; and if combined, the purpose of Beasuang's TR compiler is destroyed as discussed above in claim 1. Specifically, neither Dangelo nor Beausang teaches "prompting a user to select a memory compiler unit." Thus, the Examiner's burden of factually supporting a *prima facie* case of obviousness has clearly not been met, and the rejection under 35 U.S.C. § 103 with respect to claim 13 should be withdrawn.

Independent Claim 19

Independent claim 19 was also rejected as being unpatentable over Dangelo in view of Beausang. MPEP § 2143.03 states that "[a]ll words in a claim must be considered in judging the patentability of that claim against the prior art." (quoting *In re Wilson*, 424 F.2d 1382, 1385 (CCPA 1970)). However, in the present matter, the Examiner has not shown that all words in the claim have been considered. Specifically, independent claim 19 requires "each memory compiler unit comprises a program for assisting a multi-compiler interface" and "a program . . . to generate a combination datasheet wherein the combination datasheet comprises memory instances created by at least two of the plurality of memory compiler units." (emphasis added).

Clearly, neither Dangelo nor Beausang teach "a program for assisting a multi-compiler interface." As the Office Action points out, "Dangelo doesn't expressly disclose . . . via a multi-compiler interface." Office Action, Page 4. Also, as noted above, Beausang fails to teach a multi-compiler interface. Beausang teaching "[t]he HDL description 605 can be stored in a computer memory unit . . . and is fed into an optional generic compiler logic block 615" (column 13, lines 16–20) is not equivalent to a multi-compiler interface. Therefore, since neither Dangelo nor Beasang disclose "a multi-compiler interface," neither reference can teach "a program for assisting a multi-compiler interface."

Further, Danglo and Beausang fail to teach "the combination datasheet comprises memory instances created by at least two of the plurality of memory compiler units." The Office Action contends that Dangelo discloses "generating a combination datasheet comprising a plurality of memory instances." Office Action, Page 2. The Office Action references column 14, lines 8–15, which provides:

> [T]he data sheet generated by MemComp is used to manually extract the timing description of the memory block. This basically involves defining a set of "set.sub.-- load", "set.sub.-- drive" and "set.sub.-- arrival" constraints and associating them with the relevant pins of the surrounding logic at the start of the optimization process; or . . . a Memory Modeler (see FIG. 8) is used to generate a model 603 in Synopsys Library Language (SLL; available from LSI Logic Corporation).

Though Dangelo may teach a datasheet, Dangelo teaches that the datasheet is generated from a single memory compiler unit, not "at least two of the plurality of memory compiler units." (emphasis added). Danglo teaches "[a] Memory Compiler (MemComp) 602 takes the high level specification for memory mega-cells and produces logic and layout files for the purpose of simulation, testing, and layout." (column 13, lines 62–65). In addition, the Memory Model generates a "model 603" from the memory description: "[t]he Memory Modeler reads the memory description and generates a complete timing description of the memory block. This contains all of the setup and hold values and the timing arcs and I/O pin characteristics." (column 14, lines 16–20). Thus, the Memory Modeler does not function as a memory compiler unit or generate a combination datasheet.

Applicants respectfully submit that Dangelo teaching "the data sheet generated by MemComp" and "a Memory Modeler . . . is used to generate a model" fails to teach "a combination datasheet . . . created by at least two of the plurality of memory compiler units." Beausang fails to cure the deficiencies of Dangelo because Beausang does not teach, disclose, or suggest generating "a combination datasheet."

Accordingly, Dangelo and Beausang fail to disclose a system "wherein each memory compiler unit comprises a program for assisting a multi-compiler interface to generate a combination datasheet wherein the combination datasheet comprises memory instances created by at least two of the plurality of memory compiler units" as required by independent claim 19. Thus, the Examiner's burden of factually supporting a *prima facie* case of obviousness has

clearly not been met, and the rejection under 35 U.S.C. § 103 with respect to claim 19 should be withdrawn.

Dependent Claims 2–12, 14–18, and 20

Claims 2–12, claims 14–18, and claim 20 depend from and add additional features to and/or limit independent claims 1, 13, and 19 respectively. Therefore, these claims should be allowable for at least the reasons discussed above for the independent claims. Accordingly, Applicants respectfully request that the Examiner withdraw the rejection and allow these claims.

Conclusion

For at least the reasons set forth above, Applicants submit that the pending claims 1–20 are in condition for allowance. Accordingly, Applicants respectfully request that the Examiner withdraw the outstanding objections and rejections and issue a formal notice of allowance.

The Office Action contains characterizations of the claims and the related art to which Applicants do not necessarily agree. Unless expressly noted otherwise, Applicants decline to subscribe to any statement or characterization in the Office Action.

Please grant any extension of time required to enter this response and charge any additional required fees to our Deposit Account No. 08-1394.

Respectfully submitted,

Liem Do

Liem T. Do
Reg. No. 59,804

Dated: January 23 2008

HAYNES AND BOONE, LLP
901 Main Street, Suite 3100
Dallas, Texas 75202-3789
Telephone: (972) 739-8643
Facsimile: (972) 692-9005
Client Matter No.: 24061.122
Attorney Docket No.: 2003-0730
Document No.: R-182776_1.DOC

Certificate of Service

I hereby certify that this correspondence is being filed with the U.S. Patent and Trademark Office via EFS-Web on January 24 2008.

Bonnie Boyle

Bonnie Boyle